

Enhancing the Capabilities of QGIS Processing Toolbox

Gursimar Kaur, Sukhjot Singh Sehra, Jaiteg Singh, Hardeep Singh Rai

Department of Computer Science, Guru Nanak Dev Engineering College, Ludhiana, India

Abstract—Processing is an object-oriented Python framework for the popular open source Geographic Information System QGIS, which provides a seamless integration of geoprocessing tools from a variety of different software libraries. In this paper, we present the development history, software architecture and features of the Processing framework, which make it a versatile tool for the development of geoprocessing algorithms and workflows, as well as an efficient integration platform for algorithms from different sources. Using real-world application examples, we furthermore illustrate how the Processing architecture enables typical geoprocessing use cases in research and development, such as automating and documenting workflows, combining algorithms from different software libraries, as well as developing and integrating custom algorithms. Finally, we discuss how Processing can facilitate reproducible research and provide an outlook towards future development goals.

I. INTRODUCTION

The technological stack of Web 2.0 has enabled the voluminous production of crowdsourced data and in particular geo-spatial data. This new era of geospatial data production along with cheap hand-held electronic devices (including smartphones) has unlocked doors for innovative use (such as routing and navigation services) of this data for the common man.

The spatial data is produced specifically to fulfil the need of geographic application domains such as routing and navigation, urban planning, disaster management, land-cover analysis, etc. But, the source of data production is always been a big concern [1, 2]. National mapping agencies (NMAs) have traditionally been the official source of spatial data, which they produce and trade. Fierce utilisation of geo-tagging and spatial data in nearly every field has generated a competitive market for this data. In recent time, the private map vendors (PMV) have also become providers of more accurate and recent spatial data. PMVs mostly focus on road network data, navigation, and point-of-interests (POIs) without covering other geographic information system (GIS) application domains. PMVs are interested in producing spatial data based on products and usually avoid products that cover rural, remote areas and underdeveloped or developing countries.

The evolution of GIS research has been witnessed in the re-

cent time in diverse areas. This research area is tightly coupled with development of advances in tools and techniques to use and assess the spatial data. A shift in computing technology has also been witnessed through the literature dataset [3]. Numerous tools and custom developed codes have been used in spatial data research including proprietary tools such as ArcGIS, Manifold GIS and MapInfo [4, 5, 6, 7, 8] while open-source tools such as QGIS, JOSM, OSMOSIS, OSMIUM, PostgreSQL, and PostGIS [9, 10, 11] have been used by numerous researchers. These tools have significantly improved to handle spatial data more efficiently. Open-source databases and spatial analysis libraries have significantly improved and new functions have been added to handle spatial data more efficiently.

QGIS utilises a Python framework called processing toolbox. This toolbox provides the ability to extend the capabilities of existing algorithms [12]. The PyQGIS bindings allow comfortable integration of Python code with Qt libraries and eventually with QGIS. This toolbox provides an opportunity to develop reproducible research. Thus, the motive of this research is to discuss the methodology to enhance the geoprocessing capabilities of the Quantum GIS (QGIS). Further, methodology has been elaborated to develop the a script for assessing the data. Hence, the scripts were developed to assess and analyse the spatial data. The graphical modellers developed utilise these scripts developed in PyQGIS and are simply plug-and-play workflows.

QGIS is an Open Source Geographic Information System. The project was born in May of 2002 and was established as a project on SourceForge in June of the same year. Weve worked hard to make GIS software (which is traditionally expensive proprietary software) a viable prospect for anyone with basic access to a personal computer. QGIS currently runs on most Unix platforms, Windows, and macOS. QGIS is developed using the Qt toolkit (<https://www.qt.io>) and C++. This means that QGIS feels snappy and has a pleasing, easy-to-use graphical user interface (GUI).

QGIS aims to be a user-friendly GIS, providing common functions and features. The initial goal of the project was to provide a GIS data viewer. QGIS has reached the point in its evolution where it is being used by many for their daily GIS data-viewing needs. QGIS supports a number of raster and vector data formats, with new format support easily added using the plugin architecture.

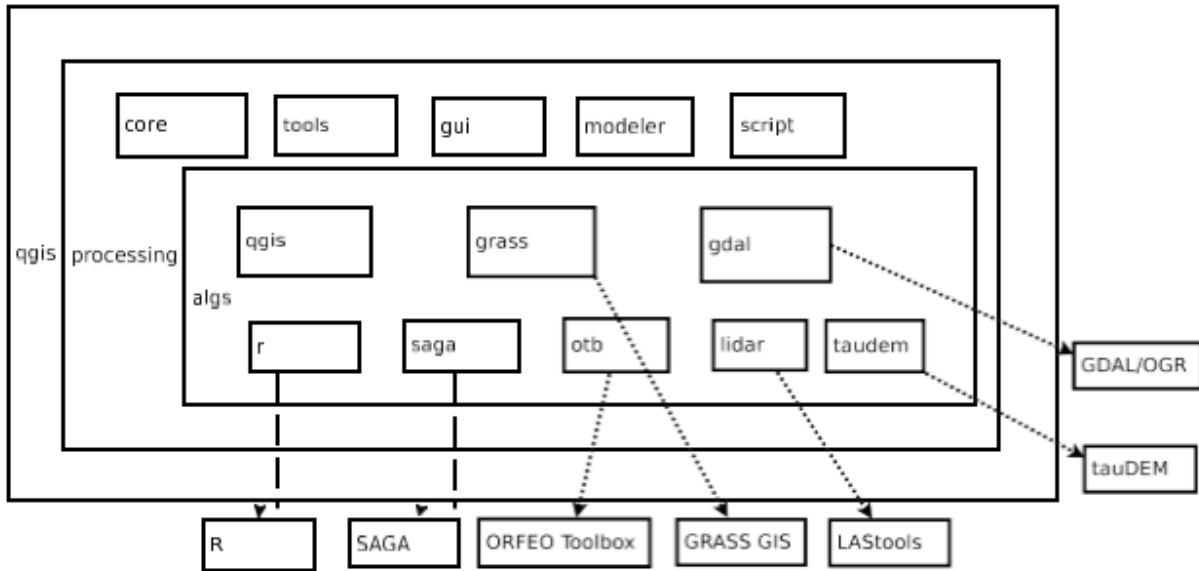


Fig. 1. QGIS Processing toolbox packages and their interaction, adapted from [12].

QGIS is released under the GNU General Public License (GPL). Developing QGIS under this license means that you can inspect and modify the source code, and guarantees that you, our happy user, will always have access to a GIS program that is free of cost and can be freely modified. You should have received a full copy of the license with your copy of QGIS, and you also can find it in Appendix GNU General Public License.

QGIS offers many common GIS functionalities provided by core features and plugins. It offers to view, explore, create, edit, manage, export, analyse data; publish maps on the Internet and extend QGIS functionality through plugins and custom developed scripts.

The current version of QGIS contains new features and extends the programmatic interface over previous versions. This version handles the issues present in the previous versions viz. interface and code reuse. The most effective and extendable part of QGIS is processing toolbox. Processing toolbox is a geoprocessing environment that can be used to call native and third-party algorithms from QGIS, making your spatial analysis tasks more productive.

The current toolbox integrates directly with multiple libraries (QGIS, GDAL/OGR, SAGA, GRASS GIS, R and ORFEO Toolbox) as shown in Figure 1 for effectively using their functions inside it. The processing toolbox provides a graphical modeller for automating the complex workflow management of algorithms. The methodology used for the assessment of OSM data was developed in Python as processing scripts. These scripts were further used as components in the

graphical modeller.

The remainder of this paper is organized as follows: Section II outlines the methodology for performing spatial analysis of geographical data using the capabilities and features of QGIS processing toolbox. Section III covers the results obtained from the algorithm used to augment the details about spatial data. Section IV draws a conclusion and directions for future work.

A. Components

II. METHODOLOGY

Taking help from the volunteer-led development project called as QGIS (Quantum GIS), standalone processing scripts were developed. This software allowed to explore the vast spectrum of analytical techniques highlighting the patterns in geographically determined data. The designed scripts can be run via Processing Framework which further offered standardized user interface. By including specifications about the input and output vector layers at the start of the script allowed to automate the entire work-flow. To perform this sort of analysis, a suitable geospatial data source is required to consider the level of accuracy between the datasets taken.

With the introduction of Web 2.0, the shift from document web to the semantic data web has changed the picture of how the internet is used by end-users [13]. The diverse and raw data that are surrounding us in various formats and structures is now linked in a relationship to build a context and shift the focus to the meaning of data. This resulted in the introduction of Semantic

Web envisioned by World Wide Web inventor Tim Berners-Lee [14]. It resulted in the creation of the universal medium for the exchange of data. Amongst the various features of DBpedia, some of the important are included as follows:

- DBpedia can be used to ask straight-forward questions over wide range of fields and topics.
- Dynamically changing according to the new additions and entries as the data in Wikipedia grows.
- Comprehensive knowledge base with free and open access for thousand of researchers and web developers.
- More than 120 language editions.
- Includes real time data

Also called as Semantic Web mirror of Wikipedia, DBpedia depends on extraction of structured information from Wikipedia. The extracted information is made available on Web and is automatically evolving. The DBpedia semantics consists of millions of 'things' and 'facts' in which a 'thing' describes a Uniform Resource Identifier (URI) and a 'fact' is used to depict Resource Description Framework (RDF) triple. RDF is a storage model which permits the storage of data files using the syntax of choice or selection. It is used to describe information conceptually. The expression of facts is achieved using three-part statements having a Subject- Predicate-Object structure. The unique feature of RDF is that unlike the traditional rules of Relational Databases, it can assign more than one value of properties to a given resource. A collection of triples is used to depict a multi-graph of undirected statements.

The prefixes in the query depicted the URI (Uniform Resource Identifier) to make the query more precise as the developer will not have to write full namespace URI over and over again. Although the developers take help of the prefixes only as preference, but it is preferably used to make the query more compact and readable. The prefixes used in the SPARQL query are: value' data in which the entire database resembles to 'Subject-Predicate-Object' triples. Algorithm II represents the query submitted over SPARQL endpoint to retrieve names

- 1) geo: < http://www.w3.org/2003/01/geo/wgs84_pos > and an object). A triple can be considered as a simple sentence
- 2) dbo: < <http://dbpedia.org/ontology/> >
- 3) owl: < <http://www.w3.org/2002/07/owl> >
- 4) foaf: < <http://xmlns.com/foaf/0.1/> >
- 5) rdfs: < <http://www.w3.org/2000/01/rdfschema> >
- 6) rdf: < <http://www.w3.org/1999/02/22rdfsyntax-ns> >
- 7) xsd: < <http://www.w3.org/2001/XMLSchema> >
- 8) bif: < *bif* : >

Taking help of popular query language to work on the Semantic Web, SPARQL allowed the users to explore structured as well as semi-structured data by querying unknown relationships in disparate database which can all be achieved in a single and simple query. It retrieved and manipulated the data stored in Resource Description Framework (RDF) format. This

using the coordinates of the feature. The dot at the last sentence signifies the end of the pattern. For example, the triple used to query for name includes foaf:name, rdfs:type, and ?name to depict the three subparts of the tripple, which are the subject, the predicate and an object. Considering the variable like an unknown character, this wildcard stored the value that matches with the position that was matching with the tripple pattern.

```
SELECT * WHERE
{
  ?place a owl:Thing;
  foaf:name ?name;
  rdfs:label ?label; geo
: geometry ?geo; geo:lat
?lat;
geo:long ?long.
FILTER (bif:stintersects(
?geo, bif:st_point(11,12), 2) &&
lang(?label) = "en"
)
```

III. RESULTS

DBpedia extracts structural and semi-structural information from Wikipedia. The information hosted on Wikipedia is open for editing and improvement of the content. Using a simple query language called as Simple Protocol and RDF query language (SPARQL) the algorithm matched the patterns to a given data source or multiple decentralized data sources and retrieved the values to obtain final answer. It is similar to SQL in its working for querying RDF data but unlike SQL, it is capable to query more than one database. Along with its new capabilities, it can unearth unknown knowledge by introducing various new features like the OPTIONAL graph, which retrieve results for the query statement only if a response exists. RDF (Resource Description Framework) expresses a fact using a triple statement of three-parts (subject, predicate,

data matches with the 'key- which is used to state a fact. Final result depended on the condition with triple patterns. The matching results obtained from the RDF graph are expressed using TURTLE syntax.

Table I represents the small dataset over which the query statement works to select all those names having value as 'NULL' or have a missing name, fetch their coordinates using the pairs of given lat/long pair to reverse code to its place name readable by end-users. The SPARQL query is used to retrieve information for a given coordinate value from DBpedia database using SPARQL endpoint and save it.

TABLE I
LABEL RESULTS FROM DBPEDIA

Id	Test dataset name
44584448	Waterlees Road
47217562	Osborne Road
47316141	First Marsh Road
176161410	Third Marsh Road
331561398	NULL
408066727	NULL

TABLE II
NAMES AND COORDINATE VALUES

Id	Name	Coordinates
331561398	Wisbech North railway station	0.149, 52.669
408066727	Leverington Hall	0.136, 52.679

It is similar to automated scraping for extraction of RDF data which depends on Wikipedia for its retrieval of results. Like Wikipedia, DBPedia is also a crowdsourced project that has participants all over the globe. Table II represents the results obtained for missing attribute names along with its coordinate values. Using the point location (latitude, longitude), the algorithm reverse coded to a readable place name for which it depended on DBPedia for enrichment of the attribute details of OSM data for making it more complete. It permitted to identify street addresses and names for adding additional information to OSM attribute details.

IV. CONCLUSION

The study proposed a simple method to explore the spatial information using the processing toolbox from QGIS framework. The selected use case required complex analysis and automation task which has been efficiently managed by the software handling activities like implementation, sharing and editing simultaneously. The features like saving and editing models in graphical modeler have made the implementation of more than one algorithms in a work-flow quick and efficient. It is observed that the people tag the attributes for map features differently due to different perceptions and mapping habits influenced by past experiences. The procedure compared the test and reference datasets using string searching algorithms. It focused on assessing the pattern with which the users trace and assign tag to the real world feature and analyzing uncertainties which prevails due to usage of semantically similar tags. As there is no prescribed guidelines on using the tagging protocol of OpenStreetMap, the users can either add a tag as considering the best fit from the selections using open suggestions, or they can create their own tag using the local knowledge and mapping experiences. The performance of the algorithms was acceptable when the corresponding features had similar geometric representation and when the computation was

made on manually generated datasets to cover varied range of diversities of how a semantically similar tags are used to assign a geographical. It further helped to assess the imbalance in spatially rich information and its attribute details in Volunteered Geographic Information as the users can assign a tag depending on various characteristics like theme, culture, local knowledge, community, linguistics and perception of real world features.

Further, the study used ever-rich database of DBPedia for querying the semantic information of the data in the geospatial information which extracts structured information from Wikipedia to retrieve results using pattern matching. On comparing the OSM data with official data, it is observed that the completeness of the crowdsourced data is acceptable. It is observed that OSM completeness is higher when computation was achieved with reference to itself, but showed variation when compared to similar but semantically related attributes. Spatial data from VGI maps is growing at a faster rate which requires extensive management of tags and rapid contribution of volunteers. The volunteers as well as the developers are required to dynamically incorporate the changes in the real world features in term of its metadata attached to the spatial data of geographical features.

REFERENCES

- [1] J. J. Arsanjani, C. Barron, M. Bakillah, and M. Helbich, "Assessing the Quality of OpenStreetMap Contributors together with their Contributions," in *16th AGILE Conference on Geographic Information Science*, Leuven, Belgium, May 2013, date- 14-17 May 2013.
- [2] T. Bartoschek and C. Keler, *VGI in Education: From K-12 to Graduate Studies*. Springer, Jan. 2013, ch. Crowdsourcing Geographic Knowledge, pp. 341–360.
- [3] H. H. Hochmair, D. Zielstra, and P. Neis, "Assessing the completeness of bicycle trail and lane features in openstreetmap for the united states," *Transactions in GIS*, vol. 19, no. 1, pp. 63–81, Feb. 2015, date- 20 February 2015.
- [4] M. Al-Bakri and D. Fairbairn, "Assessing similarity matching for possible integration of feature classifications of geospatial data from official and informal sources," *International Journal of Geographical Information Science*, vol. 26, no. 8, pp. 1437–1456, Feb. 2012, date- 10 Feb 2012.
- [5] R. Hecht, C. Kunze, and S. Hahmann, "Measuring completeness of building footprints in openstreetmap over space and time," *ISPRS - International Journal of Geo-Information*, vol. 2, no. 4, p. 1066, Nov. 2013, date- 11 November 2013. [Online]. Available: <http://www.mdpi.com/2220-9964/2/4/1066>
- [6] T. E. Chow, N. Dede-Bamfo, and K. R. Dahal, "Geographic disparity of positional errors and matching rate of residential addresses among geocoding solutions," *Annals of GIS*, vol. 22, no. 1, pp. 29–42, Oct. 2015, date- 05 Oct 2015.
- [7] S. P. Jackson, W. Mullen, P. Agouris, A. Crooks, A. Croitoru, and A. Stefanidis, "Assessing completeness and spatial error of features in volunteered geographic information," *ISPRS International Journal of Geo-Information*, vol. 2, no. 2, p. 507, Jun. 2013.

- [8] J. J. Arsanjani, A. Zipf, P. Mooney, and M. Helbich, *OpenStreetMap in GIScience*, ser. Lecture Notes in Geoinformation and Cartography. Springer International Publishing, 2015, ch. An Introduction to OpenStreetMap in Geographic Information Science: Experiences, Research, and Applications, pp. 1–15, date- 04 March 2015.
- [9] P. Neis and A. Zipf, “Analyzing the Contributor Activity of a Volunteered Geographic Information Project The Case of OpenStreetMap,” *ISPRS International Journal of Geo-Information*, vol. 1, no. 3, pp. 146–165, Jul. 2012, date- 27 July 2012. [Online]. Available: <http://www.mdpi.com/2220-9964/1/2/146/>
- [10] A. Graser, M. Straub, and M. Dragaschnig, “Towards an open source analysis toolbox for street network comparison: Indicators, tools and results of a comparison of osm and the official austrian reference graph,” *Transactions in GIS*, vol. 18, no. 4, pp. 510–526, Aug. 2014.
- [11] M. A. Brovelli, M. Minghini, M. Molinari, and P. Mooney, “Towards an automated comparison of open-streetmap with authoritative road datasets,” *Transactions in GIS*, p. Early View, Mar. 2016.
- [12] A. Graser, M. Straub, and M. Dragaschnig, *Is OSM Good Enough for Vehicle Routing A Study Comparing Street Networks in Vienna*, ser. Lecture Notes in Geoinformation and Cartography. Switzerland: Springer International Publishing, Nov. 2015, ch. Progress in Location-Based Services 2014, pp. 3–17.
- [13] M. Haklay, A. Singleton, and C. Parker, “Web mapping 2.0: The neogeography of the geoweb,” *Geography Compass*, vol. 2, no. 6, pp. 2011–2039, Nov. 2008.
- [14] A. Ballatore and M. Bertolotto, “Semantically enriching vgi in support of implicit feedback analysis,” in *Web and Wireless Geographical Information Systems*, ser. Lecture Notes in Computer Science, K.-S. K. Katsumi Tanaka, Peter Frhlich, Ed. Springer Berlin Heidelberg, Jan. 2011, vol. 6574, pp. 8–93.